Appendix A Equivalence of g-formula estimator and Snowden's g-computation algorithm

Proof that the g-formula estimation method presented in this paper is equivalent, in this setting, to the g-computation algorithm outlined by Snowden et al. (2010) for both the ATE and the ATT.

Let Y be an outcome variable and A be a binary treatment variable. Assume L is a vector of length J which represents a sufficient set for confounding adjustment. Consider the following linear model

$$E[Y|A,L] = \beta_0 + \beta_1 A + \beta_2^T L + \beta_3^T LA \tag{A.1}$$

where $\beta_2,\ \beta_3$ are parameter vectors of length J. Model A.1 is assumed to be correctly specified. The parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2^T, \hat{\beta}_3^T)^T$ are found by solving

$$\min_{\beta} \sum_i \left\{ Y_i - \left(\beta_0 + \beta_1 A_i + \beta_2^T L_i + \beta_3^T L_i A_i \right) \right\}^2 \tag{A.2}$$

where $\beta = (\beta_0, \beta_1, \beta_2^T, \beta_3^T)^T$. The Snowden g-formula estimator for the ATE is obtained using the parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2^T, \hat{\beta}_3^T)^T$ to compute the predicted values under the counterfactual scenarios of no treatment ($a = 0$) and treatment ($a = 1$) for all. Specifically, let $\hat{Y}^a = X^a \hat{\beta}$ denote the vector of predicted values for each individual under the counterfactual scenario that all individuals receive treatment a, and where the design matrix X^a has rows of the form $X_i^a = [1,\ a,\ L_i^T,\ aL_i^T]$ for $i = 1, ..., n$ and $a = 0, 1$. The Snowden g-formula estimator of the ATE can be expressed

$$\begin{aligned}
A\hat{T}E_S &= \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i^1 - \hat{Y}_i^0) \\
&= \frac{1}{n} \sum_{i=1}^{n} (X_i^1 - X_i^0) \hat{\beta} \\
&= \frac{1}{n} \sum_{i=1}^{n} \left([1,\ 1,\ L_i^T,\ L_i^T] - [1,\ 0,\ L_i^T,\ 0] \right) \hat{\beta} \\
&= \frac{1}{n} \sum_{i=1}^{n} (0,\ 1,\ 0^T,\ L_i^T) \hat{\beta} \\
&= \hat{\beta}_1 + \hat{\beta}_3^T \bar{L}
\end{aligned}$$

where $\bar{L}$ is a vector of length J with elements equal to the sample means of the J confounding variables.

Now, let $\tilde{L}_i = (L_i - \bar{L})$ and $\gamma = (\gamma_0,\ \gamma_1,\ \gamma_2^T,\ \gamma_3^T)^T$. Consider finding $\hat{\gamma} = (\hat{\gamma}_0,\ \hat{\gamma}_1,\ \hat{\gamma}_2^T,\ \hat{\gamma}_3^T)^T$ which solves

$$\min_{\gamma} \sum_i \{ Y_i - (\gamma_0 + \gamma_1 A_i + \gamma_2^T \tilde{L}_i + \gamma_3^T \tilde{L}_i A_i) \}^2$$

or equivalently

$$\min_{\gamma} \sum_i \{ Y_i - \left((\gamma_0 - \gamma_2^T \bar{L}) + (\gamma_1 - \gamma_3^T \bar{L}) A_i + \gamma_2^T L_i + \gamma_3^T L_i A_i \right) \}^2 \tag{A.3}$$

We can find $\hat{\gamma}$ by first reparameterizing. Let $\beta_0 = \gamma_0 - \gamma_2^T \bar{L}$, $\beta_1 = \gamma_1 - \gamma_3^T \bar{L}$, $\beta_2^T = \gamma_2^T$, and $\beta_3^T = \gamma_3^T$. This optimization problem is then equivalent to solving A.2 above, which yields the usual least squares estimator $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2^T, \hat{\beta}_3^T)^T$. Therefore, $\hat{\gamma}_0 = \hat{\beta}_0 + \hat{\gamma}_2^T \bar{L}$, $\hat{\gamma}_1 = \hat{\beta}_1 + \hat{\gamma}_3^T \bar{L}$, $\hat{\gamma}_2^T = \hat{\beta}_2^T$, and $\hat{\gamma}_3^T = \hat{\beta}_3^T$. That is, $\hat{\gamma}_1 = \hat{\beta}_1 + \hat{\beta}_3^T \bar{L}$. Note that $\hat{\gamma}_1$ is the estimated exposure coefficient from the g-formula model proposed in the Supplementary Methods for the ATE, and the right side of this equality is exactly $A\hat{T}E_S$ from above. The equivalence of the two g-formula estimators has thus been shown for the ATE.

The equivalence proof for the ATT estimators is analogous to the ATE proof above. Wang et al. (2017) propose an extension of Snowden et al.'s g-computation algorithm for the ATT; they show that by restricting X^a to only the treated individuals (i.e., those with $A = 1$), the algorithm returns a consistent estimate of the ATT. In particular, let $\hat{Y}^{a,\ A=1} = X^{a,\ A=1}\hat{\beta}$ denote the vector of predicted values for the n_1 individuals with $A = 1$ under the counterfactual scenario that these individuals receive treatment a, and where the design matrix $X^{a,\ A=1}$ has n_1 rows with the same form given above. Here $\hat{\beta}$ remains the usual least squares estimator found by solving A.2 above. The Wang et al. (2017) extension to the Snowden g-formula estimator for the ATT can be expressed

$$\begin{aligned}
A\hat{T}T_S &= \frac{1}{n_1}\sum_{i=1}^{n_1}(\hat{Y}_i^{1\ A=1} - \hat{Y}_i^{0,\ A=1}) \\
&= \frac{1}{n_1}\sum_{i=1}^{n_1}(X_i^{1,\ A=1} - X_i^{0,\ A=1})\hat{\beta} \\
&= \frac{1}{n_1}\sum_{i=1}^{n_1}\left([1,\ 1,\ L_i^T,\ L_i^T] - [1,\ 0,\ L_i^T,\ 0]\right)\hat{\beta} \\
&= \frac{1}{n_1}\sum_{i=1}^{n_1}(0,\ 1,\ 0^T,\ L_i^T)\hat{\beta} \\
&= \hat{\beta}_1 + \hat{\beta}_3^T\bar{L}^*
\end{aligned}$$

where $\bar{L}^*$ is a vector of length J with elements equal to the sample means among the treated of the J confounding variables.

Now let $\tilde{L}_i = (L_i - \bar{L}^*)$ and $\gamma^* = (\gamma_0^*, \gamma_1^*, \gamma_2^{*T}, \gamma_3^{*T})^T$, and consider finding $\hat{\gamma}^* = (\hat{\gamma}_0^*, \hat{\gamma}_1^*, \hat{\gamma}_2^{*T}, \hat{\gamma}_3^{*T})^T$ which solves

$$\min_{\gamma^*}\sum_i\{Y_i - (\gamma_0^* + \gamma_1^* A_i + \gamma_2^{*T}\tilde{L}_i + \gamma_3^{*T}\tilde{L}_i A_i)\}^2$$

Rearranging this optimization problem similarly to A.3 and reparameterizing analogously to above yields $\hat{\gamma}_1^* = \hat{\beta}_1 + \hat{\beta}_3^T\bar{L}^*$. Note that $\hat{\gamma}_1^*$ is the estimated exposure coefficient from the g-formula model proposed in the Supplementary Methods for the ATT, and the right side of this equality is exactly $A\hat{T}T_S$ from above. Thus the equivalence of the two g-formula estimators has been shown for the ATT as well.

The proof for the equivalence of the average treatment effect in the untreated (ATU) is similar to the proof for the ATT.

Appendix B R Workflows

B.1 Generate Population Data and Analysis Dataset

This section contains R code for (i) generating a simulated dataset similar to those used in the simulation studies of the main text, and (ii) analyzing these data with each of regression, IPW, and the parametric g-formula to get the estimated effect of smoking in the smokers and the corresponding estimated standard errors. The dataset contains 770 individuals, which are sampled from a population of 10 million people.

```r
set.seed(1)
n <- 770     # individuals per dset
nsim <- 1    # number of simulated dsets
nsup <- 1e7  # population size
```

The following variables are generated for the population, according to the observed characteristics of the METSIM cohort.

```r
## Age
age <- rnorm(nsup, mean=54.76, sd=5.07)
age.c <- scale(age, scale = TRUE)

## Alcohol consumption
totalcw <- rexp(nsup, rate=1/104.6)
totalcw.c <- scale(sqrt(totalcw), scale = TRUE)

## Everyday vegetable consumption
veg <- rbinom(nsup, 1, 0.83)

## Hobby exercise level
hex.p <- cbind(0.05 - 0.01*veg, 0.28 - 0.03*veg, 0.18 + 0.01*veg,
               0.49 + 0.03*veg)
hobbyex <- sapply(1:nsup, function(i) sample(1:4, 1, replace=TRUE,
                                             prob=hex.p[i,]))

## Body Mass Index
BMI.mn <- 26.59 - 0.33*(veg-0.5) - 0.25*(hobbyex - 1)
BMI <- rnorm(nsup, mean=BMI.mn, sd=3.47)
BMI.c <- scale(BMI, scale = TRUE)

## Current smoking
lp.smk <- exp(0.51 - 0.66*veg + 0.51*totalcw.c -
                0.57*hobbyex - 0.42*BMI.c)
smoke.cu <- rbinom(nsup, 1, prob=lp.smk/(1+lp.smk))

dsn <- matrix(data=cbind(age, age.c, totalcw, totalcw.c, veg,
                         hobbyex, BMI, BMI.c, smoke.cu,
                         smoke.cu*totalcw.c, smoke.cu*veg,
                         smoke.cu*hobbyex, smoke.cu*BMI.c),
              nrow=nsup)
colnames(dsn) <- c("age", "age.c", "totalcw", "totalcw.c", "veg",
                   "hobbyex", "BMI", "BMI.c", "smoke.cu",
                   "smk_totalcw", "smk_veg",
                   "smk_hobbyex", "smk_BMI")
```

The dataset is assembled by sampling without replacement from the population. Note that if multiple datasets were being generated (i.e., nsim>1), the same individual may be represented in more than one dataset. However, as written, this code doesn't allow for individuals to be represented more than once within a single dataset.

```
samps <- matrix(sapply(1:nsim,
                       function(x) sample(1:nsup,
                                          n,
                                          replace=FALSE)),
                nrow=nsim, ncol=n, byrow = TRUE)
dsn.s <- matrix(dsn[t(samps),], nrow=nsim*n, ncol=ncol(dsn))
colnames(dsn.s) <- colnames(dsn)
```

B.2 Generate Gene Expression Data

Next, a matrix of coefficients is defined to set the influence of each of the covariates, the exposure, and their interactions on the average treatment effect in the treated. These six sets of coefficients correspond to the six simulation scenarios described in the main text.

```
## Coefficients of exposure, covariate, and intxn terms
## for individual gene expression means
coef <- matrix(c(rep(0, 10),
                 c(0.05, -0.1, 0.15, -0.2, 0.25, -2, rep(0, 4)),
                 c(-0.5, 1.0, -1.5, -1.0, 0.5, 2, rep(0, 4)),
                 c(rep(0.05, 5), 2, -1, 0, 0, 0),
                 c(rep(0.05, 5), 2, 1, 0, -1, 0),
                 c(rep(0.05, 5), 2, 0, 0, -1.5, -1.5)), nrow=10)
colnames(coef) <- c("Null", "NoInt1", "NoInt2", "Int1", "Int2", "Int3")
rownames(coef) <- c("age.c", "totalcw.c", "veg", "hobbyex", "BMI.c",
                    "smoke.cu", "smk_totalcw", "smk_veg",
                    "smk_hobbyex", "smk_BMI")
ngn <- ncol(coef)
```

The individual covariate values are combined with the coefficients to produce a mean expression level for each gene in each individual. In other words, each person is given a mean expression level for each of the simulated genes based on their set of covariate and exposure values. These means are used to generate gene expression values from a normal distribution with standard deviation 0.24, for each gene.

```
## Compute the mean expression level for each simulated gene
covars.gnmn <- rownames(coef)
gn.mn.s <- dsn.s[ ,covars.gnmn] %*% coef

GE <- matrix(sapply(1:ngn,
                    function(x) rnorm(nsim*n,
                                      mean=gn.mn.s[,x],
                                      sd=0.24)),
             nrow=n*nsim, ncol=ngn, byrow=FALSE)
colnames(GE) <- colnames(coef)
```

B.3 Get True ATT for Each Gene

The true population ATT is computed for each gene by summing the contributions from each covariate. The covariate main effects don't contribute to the ATT, only the exposure and the exposure-covariate interaction terms. Since the ATT is of interest, the contributions made by the interaction terms are the product of the coefficients and the mean value of each covariate in the current smokers (i.e., in the exposed). The contribution from the exposure is just the coefficients themselves.

```
dsn <- data.frame(dsn)
covmn.trt <- apply(dsn[which(dsn$smoke.cu == 1),
                       c("totalcw.c", "veg", "hobbyex", "BMI.c")],
                   2, mean)
smk.eff <- coef
smk.eff[1:5,] <- 0
smk.eff[7:10,] <- covmn.trt*smk.eff[7:10,]

tru.att <- colSums(smk.eff)
```

B.4 Analyze Simulated Dataset

Analyses are conducted according to the models used in the main text for each method. This includes the addition of the quadratic term for age, though in simulations this term doesn't contribute to the ATT. For each method, the estimates and their estimated standard errors can be used to construct Wald confidence intervals and get t-statistics and p-values.

```
dsn.s <- data.frame(dsn.s)
dsn.s$age2 <- I(dsn.s$age^2)
```

B.4.1 Traditional Regression Analysis

The linear regression model is fit to the data once per gene using the `lmFit` function from the `limma` package for convenience and speed. Note that the standard errors recorded are the usual least squares estimates, not the moderated standard errors that the `lmFit` function also produces.

```
## Create design matrix and fit linear model for each gene
design <- model.matrix(~smoke.cu + BMI.c + veg + totalcw.c +
                         hobbyex + age.c + scale(age2), dsn.s)
fit <- lmFit(t(GE), design)

## Collect and display results
Ests <- fit$coefficients[,2]
SEs <- fit$sigma*fit$stdev.unscaled[,2]
data.frame(rbind(Ests, SEs))
```

```
##             Null     NoInt1     NoInt2       Int1        Int2        Int3
## Ests 0.002323104 -2.0439902 2.01500677 1.74938784 -0.67929583 -1.98868044
## SEs  0.025139323  0.0245301 0.02507283 0.04947473  0.05932175  0.07061574
```

B.4.2 Inverse Probability Weighting Analysis

Begin the IPW analysis by fitting a logistic regression model of the exposure to obtain parameter estimates needed for constructing the IP weights. Once the weights have been computed, it is good practice to check their distribution for extreme values and to ensure the mean is close to its expected value $E[W^{ATT}] = 2P(A = 1)$. For this dataset, there appear to be no extreme values and the mean is very close to its expected value.

```
fit_wts <- glm(smoke.cu ~ BMI + veg + totalcw + hobbyex + age + age2,
               family=binomial(link="logit"), data=dsn.s)
ncov <- length(coef(fit_wts))
wt.sato <- ifelse(dsn.s$smoke.cu == 0,
                  exp(fit_wts$linear.predictors), 1)
summary(wt.sato)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.01905 0.10330 0.17826 0.33902 0.42134 1.28176
```

```
2*mean(dsn$smoke.cu) # 2*P(A=1)
```

```
## [1] 0.3317606
```

A simple linear regression model is fit for each gene with the weights, again using the `lmFit` function. The estimated ATT for each gene is recorded.

```
## Fit simple linear reg model
design <- model.matrix(~smoke.cu, dsn.s)
fit <- lmFit(t(GE), design, weights=wt.sato)
Ests <- fit$coefficients[,2]
```

The standard error estimates from ordinary least squares are not appropriate in general for the IPW estimator, so the `geex` package is used to compute consistent standard error estimates. This approach stacks the estimating equations for the weights with those for the Hajek estimator using the function `estfun_IPW`, thereby accounting for estimation of the weights. The `m_estimate` function of the `geex` package computes the variance-covariance matrix for all estimated quantities, and the elements corresponding to the Hajek estimator are selected to compute the standard error estimate for each gene.

```
## First argument required for m_estimate function of geex
## Constructs the stacked estimating equations
estfun_IPW <- function(data, model){
  L <- model.matrix(model, data=data)
  A <- model.response(model.frame(model, data=data))
  Y <- data[,1]

  function(theta){
    p  <- length(theta)
    p1 <- length(coef(model))
    lp  <- L %*% theta[1:p1]
    rho <- plogis(lp)

    IPW <- ifelse(A == 1, 1, exp(lp))

    score_eqns <- apply(L, 2, function(x) sum((A - rho) * x))
    ce1 <- IPW*(A==1)*(Y - theta[p-1])
    ce0 <- IPW*(A==0)*(Y - theta[p])

    c(score_eqns,
      ce1,
      ce0)
  }
}

## Estimated counterfactual means
notexp <- unname(which(fit$design[,2] == 0))
mu0_hat <- rowMeans(fitted(fit)[,notexp])
mu1_hat <- rowMeans(fitted(fit)[,-notexp])

## Compute SEs, accounting for weight estimation
vcovs <- sapply(colnames(GE),
                function(x) vcov(m_estimate(
                  estFUN = estfun_IPW,
                  data  = data.frame(GE[,x],
                                     dsn.s[,c("age", "age2",
                                              "totalcw", "veg",
                                              "BMI", "hobbyex",
                                              "smoke.cu")]),
                  roots = c(coef(fit_wts), mu1_hat[x], mu0_hat[x]),
                  compute_roots = FALSE,
                  outer_args = list(model = fit_wts)
                )), simplify = FALSE  )
SEs <- sapply(1:ncol(GE),
              function(x) sqrt(vcovs[[x]][ncov+1, ncov+1] +
                               vcovs[[x]][ncov+2, ncov+2] -
                               2*vcovs[[x]][ncov+1, ncov+2]))
data.frame(rbind(Ests, SEs))
```

```
##             Null      NoInt1     NoInt2      Int1       Int2       Int3
## Ests 0.007559031 -2.04134088 2.07096587 1.6588133 -0.5233402 -1.8802092
## SEs  0.026950887  0.02623274 0.06423069 0.1077299  0.1346590  0.1568875
```

B.4.3 Parametric G-formula Analysis

Fitting the g-formula model requires first centering all non-exposure covariates at their sample mean in the exposed. Once the appropriately centered main effects and interactions are generated, the full regression model is fit to obtain the estimated ATT for each gene, once more using lmFit. As with IPW, the standard error estimates are computed using stacked estimating equations to take the estimation of the covariate means into account.

```
## Covariate means in the exposed
pdat <- dsn.s[, c("smoke.cu", "BMI", "veg", "totalcw",
                  "hobbyex", "age", "age2")]
covs <- pdat[, -1]
cov.mns <- apply(covs[which(pdat$smoke.cu == 1),], 2, mean)

## Create centered covariates
ncov <- length(cov.mns)
pdat[, c("BMI.c", "veg.c", "tot.c", "hob.c", "age.c", "age2.c")] <-
  sapply(1:ncov, function(x) covs[,x] - cov.mns[x])

## Create intxn terms
intx <- c("smkBMI", "smkveg", "smktot", "smkhob")
cov.t <- pdat[, c("BMI.c", "veg.c", "tot.c", "hob.c")]
pdat[, intx] <- pdat$smoke.cu * cov.t

## Fit reg model
design <- model.matrix(~smoke.cu + BMI.c + veg.c + tot.c + hob.c +
                         age.c + age2.c + smkBMI + smkveg + smktot +
                         smkhob, pdat)
fit <- lmFit(t(GE), design)
Ests <- fit$coefficients[,2]
```

Similarly to above, the estfun_gf function is required by geex to produce the set of stacked estimating equations. The m_estimate code is analogous to that shown above for IPW, but here the standard error estimate of the g-formula estimator can be obtained directly from the diagonal of the variance-covariance matrix for each gene.

```r
## First argument required for m_estimate function of geex
## Constructs the stacked estimating equations
estfun_gf <- function(data){
  L <- data[,3:ncol(data)]
  A <- data[,2]
  Y <- data[,1]
  I <- rep(1, length(Y))

  function(theta){
    pL <- ncol(L)
    X.Lc <- matrix(NA, nrow = length(Y), ncol = pL)
    for (i in 1:pL) {
      X.Lc[,i] <- L[,i] - theta[i]
    }

    X <- cbind(I, A, X.Lc,
               A*X.Lc[,1], A*X.Lc[,2], A*X.Lc[,3], A*X.Lc[,4])
    p <- ncol(X)

    b <- cbind(theta[(pL+1):(p+pL)])
    Xb <- X %*% b

    mn_eqns <- apply(X.Lc, 2, function(x) sum(A * x))
    score_eqns <- apply(X, 2, function(x) sum((Y - Xb) * x))

    c(mn_eqns,
      score_eqns)
  }
}

## SEs accounting for weight estimation
SEs <- sapply(colnames(GE),
              function(x) sqrt( vcov(m_estimate(
                estFUN = estfun_gf,
                data  = data.frame(GE[,x],
                                   pdat[,c("smoke.cu", "BMI",
                                           "veg", "totalcw",
                                           "hobbyex", "age",
                                           "age2")]),
                roots = c(cov.mns, fit$coefficients[x,]),
                compute_roots = FALSE
              ))[ncov+2, ncov+2] ) )

data.frame(rbind(Ests, SEs))
```

```
##             Null      NoInt1     NoInt2      Int1       Int2       Int3
## Ests 0.004852312 -2.03990409 1.93530314 1.6513069 -0.5340888 -1.8797919
## SEs  0.027014533  0.02551567 0.04884979 0.1090459  0.1330511  0.1560926
```

Supplementary Methods

Data Preprocessing

Microarray data (Affymetrix Human Genome U219 Array) from the METSIM project (Laakso et al., 2017) was downloaded from GEO accession GSE70353 (Civelek et al., 2017) using the Bioconductor package GEOquery (Davis and Meltzer, 2007). The downloaded data was normalized by the study authors. Microarray measurements per probeset were summarized for each gene using the median polish method from Tukey (1977) (the `medpolish` function in the R programming environment) on $\log_2$ transformed, normalized expression data.

In general, missing covariate values in the data set must be addressed before employing regression, IPW, or the parametric g-formula. If the percentage of individuals with missing covariates is low and the data are believed to be missing completely at random (MCAR), then a complete case analysis may be expected to not introduce bias. However, it is rarely the case that both of these criteria are met, and so it is recommended to take a more sophisticated approach such as multiple imputation (Moodie et al., 2008; Perkins et al., 2018). For further details and recommendations on handling missing covariates data when fitting causal models, see Moodie et al. (2008).

Assume for all of the following models that expression level for gene g has been $\log_2$ transformed and normalized, and that all probe sets have been collapsed (e.g., using median polish) resulting in one measure per gene per subject. All vectors are assumed to be column vectors throughout.

Comparing Approaches for Exposure Effect Estimation

Let Y_g represent the observed expression level for gene g and A be a binary exposure of interest taking on values 0 or 1. For $a = 0,1$, let Y_g^a denote the expression level for gene g had, possibly counter to fact, the exposure level been a. These Y_g^a are often referred to as counterfactuals or potential outcomes, and only at most one of the potential outcomes Y_g^1 and Y_g^0 for gene g are observed for any given individual. The target of inference here is the ATT for gene g and is defined as $ATT_g = E[Y_g^1 - Y_g^0|A = 1]$, the average effect of the exposure on expression for gene g in the population of exposed individuals. The ATE for gene g is similarly defined as $ATE_g = E[Y_g^1 - Y_g^0]$, the average effect of the exposure on expression for gene g in the population of all individuals. The methods outlined below use the observed data to estimate exposure effects on gene expression, and the assumptions sufficient for valid inferences on ATT_g for each of these methods are parsed out further in the remaining sections of the Supplementary Methods.

Regression

For modeling observed expression level Y_g of gene g as a function of some exposure of interest A in the presence of confounding, linear regression is the conventional approach. Confounders L, where L is a vector of length J, were included in the model as covariates along with the exposure variable A. In particular, the model can be written

$$E[Y_g|A,L] = \theta_{g0} + \theta_{g1}A + \theta_{g2}^T L \tag{S.1}$$

for each gene g, where θ_{g2} is also a vector of length J. The estimated exposure effect $\hat{\theta}_{g1}$ and its estimated standard error were computed in the usual fashion using ordinary least squares (OLS). The estimator $\hat{\theta}_{g1}$ in S.1 is interpretable as estimating the ATE for gene g, which does not necessarily equal the ATT for gene g, unless operating under the assumption $ATE_g = ATT_g$.

The omission of interaction terms from model S.1 assumes the exposure and confounders are not interacting to influence gene expression, which is not necessarily true in general. Therefore if any such interactions are present in the data, this regression model will not account for those relationships and can then yield biased exposure effect estimates. On the other hand, as noted in the Discussion section, the inclusion of interaction terms yields conditional exposure effects. Specifically, there is no one parameter in the regression model with interactions that is interpretable as the ATT. If exposure-confounder interactions are included in model

S.1, then the estimated coefficient of the exposure variable estimates the exposure effect for an individual with confounder variable values all equal to zero. This quantity does not describe the average exposure effect in the population, and depending on the range of values for the included confounding variables, it may not even be close to describing the exposure effect for any people in the population. For these reasons, neither the regression model with nor without interactions will yield consistent effect estimates in general. The one exception where regression with interactions can consistently estimate the ATT is when all confounders are centered at the mean in the treated, which is exactly the g-formula; for further details see the Parametric g-formula section below and Appendix A. Since this manuscript is focused on marginal exposure effects, interactions were omitted from the regression model in the main text. However, results of fitting the regression model with interactions are included later in the Supplementary Results section for both the simulated and METSIM data.

Inverse Probability Weighting

Inverse probability weights were computed by fitting the following logistic regression model with the binary exposure A as the outcome and the set of J confounders L as the predictors:

$$logit(P(A=1|L)) = \alpha_0 + \alpha_1^T L \tag{S.2}$$

where α_1 is a vector of length J. The fitted values from this logistic regression model were used to construct the individual weights that were then used in the models for gene expression. Choice of weights depends on the target population; this paper focuses on the ATT, so the weights given below first derived in Sato and Matsuyama (2003) were used. These weights take the form for each individual i of the ratio of the conditional probability of the subject being exposed to the conditional probability of the subject's actual exposure status. That is, the weight for subject i equals

$$W_i^{ATT} = A_i + (1-A_i)\exp(\alpha_0 + \alpha_1^T L_i) \tag{S.3}$$

So, if a subject was exposed their weight was simply equal to one. For unexposed subjects, weights were estimated by substituting in the estimates $\hat{\alpha}_0$, $\hat{\alpha}_1$ from fitting model S.2 for the true parameter values in S.3, i.e., $\hat{W}_i^{ATT} = A_i + (1-A_i)\exp(\hat{\alpha}_0 + \hat{\alpha}_1^T L_i)$. When using IPW it is good practice to check that the mean of the weights is close to their expected value; for these weights, $E[W_i^{ATT}] = 2P(A=1)$.

After estimating the weights, the linear regression model

$$E[Y_g|A] = \theta_{g0} + \theta_{g1}A \tag{S.4}$$

was fit for each gene g using weighted least squares with weights $\hat{W}_i^{ATT}$, yielding

$$\hat{\theta}_{g1} = \frac{\sum_{i=1}^n \hat{W}_i^{ATT} A_i Y_{gi}}{\sum_{i=1}^n \hat{W}_i^{ATT} A_i} - \frac{\sum_{i=1}^n \hat{W}_i^{ATT}(1-A_i) Y_{gi}}{\sum_{i=1}^n \hat{W}_i^{ATT}(1-A_i)}$$

This estimator is sometimes referred to as the Hajek or modified Horwitz-Thompson estimator (Hernán and Robins, 2020), and is consistent for ATT_g. Note that consistency of the Hajek estimator depends on the model for $A|L$ in S.2 being correctly specified. No outcome model is assumed; fitting S.4 by weighted least squares is simply a convenient way to compute the Hajek estimator using standard software.

Parametric g-formula

In this final approach, the following altered version of the initial linear regression model was fit to the data.

$$E[Y_g|A,\tilde{L}] = \theta_{g0} + \theta_{g1}A + \theta_{g2}^T\tilde{L} + \theta_{g3}^T\tilde{L}A \tag{S.5}$$

where $\tilde{L} = (L-\tau)$ and τ is a vector of constants of length J, and θ_{g2}, θ_{g3} are parameter vectors of length J. This model differs from model S.1 in two key ways: all first order interactions between A and the covariates L have been added, and all covariates L have been centered at τ. Since the ATT was of primary interest here, τ was chosen to equal the means of the covariates L among the exposed, $E[L|A=1]$. Since the true means are unknown, a consistent estimate for τ was substituted, namely $\hat{\tau} = \sum_i A_i L_i / \sum_i A_i$. If interested in the ATE instead, let $\tau = E[L]$ and consistently estimate using $\hat{\tau} = \sum_i L_i / n$.

The model was then fit using OLS to obtain the estimated ATT_g, $\hat{\theta}_{g1}$. This estimator is equivalent to the Snowden et al. (2010) estimator and is consistent for ATT_g; see Appendix A.

Standard Error Estimators for Each Method

The standard error for the exposure effect estimator from the linear regression model was obtained using the estimated variance matrix resulting from fitting the model with OLS, in keeping with the conventional approach. Estimating equations were used to compute the standard error of both the IPW Hajek estimator and the parametric g-formula estimator (Stefanski and Boos, 2002).

When taking an estimating equations approach to computing the standard errors, for IPW one must decide whether or not to take the estimation of the weights into account. The variance estimator that results from treating the weights as known is referred to here as the robust sandwich variance estimator (robust SVE). Computing the robust SVE is readily accomplished using various R packages such as *sandwich* or *geepack*. Accounting for the weight estimation in the variance computation, on the other hand, can be accomplished through supplying the set of estimating equations to the *geex* package (Saul and Hudgens, 2020) in R.

When using IPW and the ATE is of interest, the robust SVE is conservative when the weights are assumed known and consistent when weight estimation is taken into account (Lunceford and Davidian, 2004). If using IPW to estimate the ATT, it is known from the theory of M-estimation (Stefanski and Boos, 2002) that this variance estimator is consistent when weight estimation is taken into account; however, it can be either conservative or anti-conservative when weights are assumed fixed. If computing the standard errors with *geex*, the set of estimating equations needed include the score equations from the logistic regression model in S.2 along with the two estimating equations corresponding to the two pieces of the Hajek estimator.

Bootstrapping is commonly employed to estimate standard errors when the g-formula is used to estimate the ATE or ATT (Efron and Tibshirani, 1986; Snowden et al., 2010; Wang et al., 2017), which is a valid option, but using stacked estimating equations provides a closed-form alternative. It is recommended when using the estimating equations approach that the covariate mean estimation be taken into account; again, by estimating equation theory, these standard errors are consistent and yield valid confidence intervals. If using *geex* to compute the standard errors for this estimator, the set of estimating equations needed are those corresponding to the estimation of each covariate mean and the parameters in model S.5.

In the simulations and data analyses of the paper, the variance of the IPW estimator for ATT_g was estimated both ways and the two estimates were found to be fairly different in some instances and nearly identical in others. The same approach was taken for the variance of the g-formula estimator for ATT_g, and the standard errors were substantially larger when accounting for estimation of covariate means. The standard error estimates reported were computed taking into account the estimation of the weights and the covariate means. The R markdown workflow that accompanies this paper includes code for computing the variance using stacked estimating equations for both IPW and the g-formula.

The standard errors for all methods were used to construct Wald 95% confidence intervals and perform t-tests of $H_0 : \theta_{g1} = 0$, i.e., no effect of exposure on gene expression in the exposed for gene g.

Assumptions

With all methods presented here, it is assumed the gene expression data have already been normalized and reduced to one observation per person per gene (i.e., not probe-level data). These methods also require that there are no missing values; if missing values are present in the covariates L or exposure A, see the recommendations in the section above. For these methods to yield consistent estimates, it is also assumed that there is no bias due to selection or systematic measurement error. Importantly, formal arguments for IPW and the parametric g-formula estimators involve asymptotic justifications, and there is no guarantee that these methods will perform well for small or moderate sample sizes (e.g., $n < 40$ for IPW (Pirracchio et al., 2012)).

In order for the IPW and g-formula methods to adequately adjust for confounding, the set of covariates L must satisfy the conditional exchangeability assumption ($Y_g^a \perp\!\!\!\perp A|L$). It is also required that positivity $P(A = a|L = l) > 0$ for all l where $dF_L(l) > 0$ and F_L is the CDF of L, and the Stable Unit Treatment Value Assumption (SUTVA) hold. SUTVA requires causal consistency, i.e., no different versions of exposure, and no interference, i.e., one individual's exposure status doesn't affect another individual's gene expression.

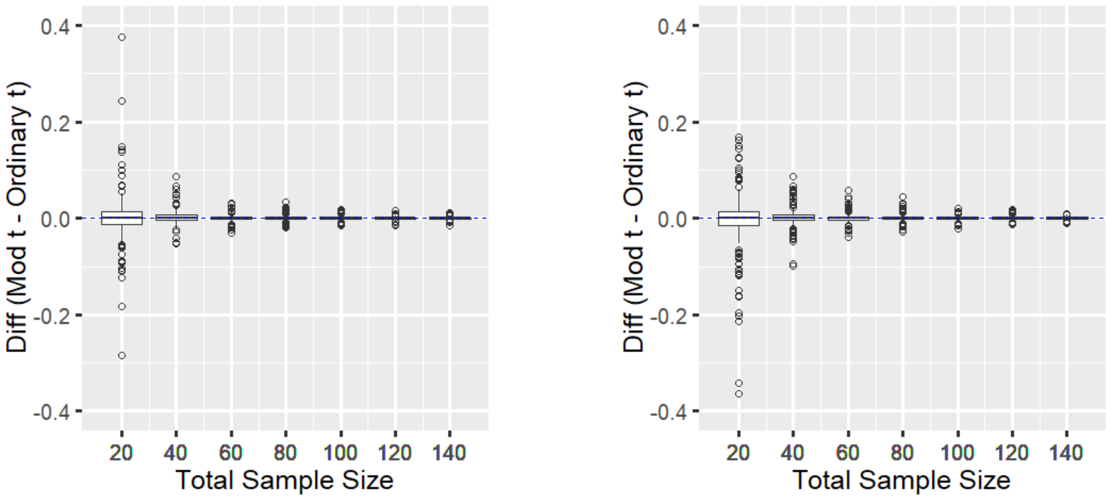
For the IPW and g-formula estimators to yield consistent effect estimates, the above assumptions must hold. When using IPW, the additional assumption that the model of $A|L$ is correctly specified is needed as well. For the g-formula, no specification of a model for $A|L$ is needed, but the model for $Y_g|A,L$ must be correctly specified.

Supplementary Results

Simulation Study for Ordinary vs Moderated *t* Statistic

In observational genomics studies, the total sample size is often large enough for results depending on large sample theory to hold. The *limma* package (Smyth, 2004) is used in standard practice to obtain effect estimates, *t*-statistics, and p-values for each gene when assessing the effect of some exposure on gene expression. This package computes a moderated *t*-statistic that is shown to perform well in small sample sizes, as are found in traditional genetic studies, and which converges to the ordinary *t*-statistic as the sample size increases. The moderation of this *t*-statistic comes into play with empirical Bayes moderation of the standard errors toward a common value; here it is shown empirically that these moderated standard errors are practically equivalent to the ordinary standard errors in large samples.

In Figure S.1 the difference between the moderated and ordinary *t*-statistics for an example gene are given for a variety of sample sizes, and with balanced and unbalanced group assignment. In particular, a randomly chosen gene from the METSIM cohort data was used, and the 770 participants were randomly sampled according to their current smoking status to create the analysis datasets. For each combination of sample size and group allocation, 200 analysis datasets were constructed. Samples were drawn such that the same individual may have been represented in more than one dataset, but not more than once within a single dataset. Both types of *t*-statistic and their difference were computed for each dataset, and boxplots of the 200 differences are given for each scenario in Figure S.1. Regardless of the group allocation, once the total sample size was around 100 or larger, the moderated and ordinary *t*-statistics were practically equivalent.



(a) Balanced group sizes (1:1)

(b) Unbalanced group sizes (5:1)

Supplementary Materials, Figure S.1: Difference between the moderated and ordinary *t*-statistics for various sample sizes with (a) balanced (1 non-smoker : 1 smoker) and (b) unbalanced groups (5 non-smokers : 1 smoker). The boxplot for each sample size represented in (a) and (b) summarizes the difference in *t*-statistics for 200 datasets, each derived from the same gene.

All analyses in this paper were conducted with well over 100 individuals, and so ordinary linear regression was used for simplicity.

Analyses Using Regression Model with Interactions

In the main text, the traditional regression model was constructed without any interaction terms. This section presents results of the simulation studies designed in the main text and the METSIM analysis using the regression approach with interactions in the model. Specifically, interactions between smoking (*smk*) and each of alcohol consumption (*alc*), vegetable consumption (*veg*), hobby exercise (*hex*), and BMI (*bmi*) were included. These terms were chosen because they reflect the interactions included in the parametric g-formula approach. Both *alc* and *bmi* were centered at their population mean in the main effects and the respective interaction terms to avoid collinearity with the intercept, as was done in the traditional regression analysis.

The empirical bias and confidence interval coverage and width for the ATT estimator from the regression model with interactions are given in Table S.1 below, again averaging over 1000 simulations per scenario. The simulated data sets used were identical to those that were generated and analyzed in the main text. Comparing the results in Table S.1 to those in Table 3, it is evident that both regression estimators showed bias and failed to meet nominal coverage in the presence of exposure-covariate interactions. That is, the addition of the interaction terms to the regression model did not counteract the estimator bias for these scenarios, and in fact worsened the bias for multiple cases. The regression with interactions approach performed well in terms of estimator bias and CI coverage for the first three scenarios, but the average CI width in these cases greatly exceeded that for the other methods. Further, the regression model with interactions doesn't generally produce consistent marginal exposure effect estimates; the exposure effects estimated from this model were conditional on the particular values of the confounding variables.

The exposure effect from the regression model with interactions, as reported here, is interpreted as the effect of smoking on gene expression for an individual with alcohol consumption and BMI equal to the mean in the sample, who doesn't consume vegetables everyday, and who has an undefined hobby exercise level (i.e., *hex* = 0). As mentioned in the Supplementary Methods, this exposure effect estimate doesn't necessarily describe any observed individual in the population. Ultimately, obtaining one marginal estimate to describe a population is desirable for many researchers, and the conditional estimates produced by the model with interactions are not readily combined to produce a marginal estimate. There are multiple confounding variables, some of which are continuous, resulting in an unwieldy number of conditional estimates to work with.

Supplementary Materials, Table S.1: Average empirical bias, 95% confidence interval coverage, and average width for the regression with interactions estimator.

Scenario	True ATT	Estimate Bias	95% CI Coverage	95% CI Width
Null Case	0.00	0.00	0.95	0.33
No Interactions 1	-2.00	-0.01	0.95	0.33
No Interactions 2	2.00	0.06	0.93	0.51
Interactions 1	1.59	0.36	0.09	0.36
Interactions 2	-0.36	2.41	0.00	0.37
Interactions 3	-1.75	3.75	0.00	0.33

When fit using the METSIM data, the regression model with interactions returned the same top two genes as ranked by p-value, but the smoking effect estimates and SEs were strikingly different from the other estimators. In particular, the smoking effect estimates for the top two genes, *CYP1A1* and *CYP1B1* were, respectively, 2.81 and 0.99. Both smoking effect estimates were substantially larger than the estimates produced by the other three methods (from Figure 1: approximately 2.1 and 0.75 for *CYP1A1* and *CYP1B1*, respectively). The SE estimates for these top two genes (0.20 and 0.16, respectively) were approximately three times larger than the SE estimates produced by the regression model without interactions. This result is in accordance with those shown above for the simulation studies.

Root Mean Square Error for Simulation Study Results

To further examine the bias-variance trade-off of the ATT estimators, the root mean square error (RMSE) was calculated over the 1000 simulated datasets from the main text and the results are given in Table S.2. In every scenario considered, the estimator from the regression model with interactions had the highest RMSE; this is not surprising given the bias and CI width results presented in Table S.1. With the exception of the Interactions 3 scenario, the RMSE for the IPW and g-formula estimators were very close. The only scenario considered here where the estimator from the traditional regression model had RMSE clearly lower than both IPW and the g-formula was Interactions 3, but the g-formula RMSE was only slightly larger.

Supplementary Materials, Table S.2: Root Mean Square Error for the regression, IPW, parametric g-formula, and regression with interactions ATT estimators.

Scenario	Reg	IPW	G-form	Reg + Int
Null Case	0.024	0.027	0.025	0.083
No Interactions 1	0.138	0.099	0.098	0.381
No Interactions 2	0.229	0.128	0.128	2.412
Interactions 1	0.253	0.177	0.178	3.754
Interactions 2	0.025	0.031	0.025	0.086
Interactions 3	0.045	0.136	0.050	0.144

Sensitivity Analyses of METSIM Microarray Data

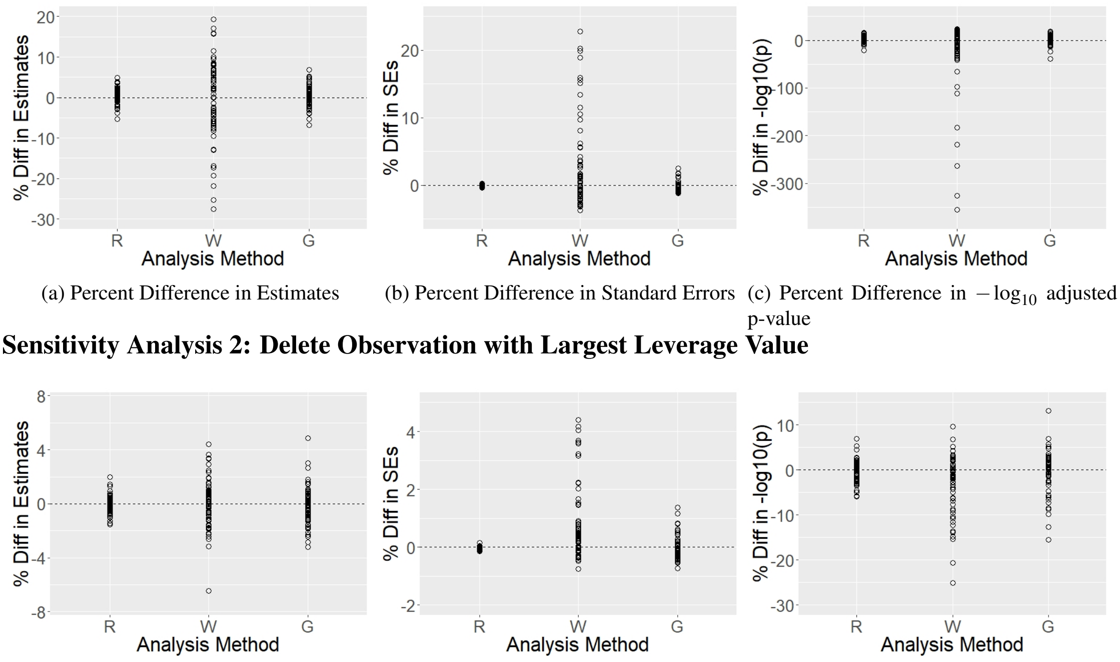
As mentioned in the main text, there was one large IP weight from the METSIM primary analysis whose influence deserves further investigation. Deleting this individual from the data resulted in a sample of size 769, which was analyzed again in the same manner as above. Results of the regression, IPW, and g-formula sensitivity analysis are compared to the results from the primary analysis in Figure S.2a - S.2c.

Additionally, the leverage values for each individual were computed from the design matrices of the g-formula and regression methods. The individual with highest leverage value was the same for both methods, and another sensitivity analysis was performed by deleting this individual; the results of this second sensitivity analysis are compared to the results from the primary analysis in Figure S.2d - S.2f. Notably, the individual with second highest leverage value was the same individual who generated the largest weight and who was deleted in the sensitivity analysis above. The genes represented in this figure are the same set as those in Figure 1, namely the top 50 genes as ranked in the primary analysis.

In particular, the first row of this figure shows the deletion of the observation with largest weight had very little effect on the regression and g-formula estimates, standard errors, and p-values. On the other hand IPW appears to be more sensitive to the deletion of this observation, with percent difference in effect estimates and standard errors ranging up to a magnitude of 30 and 25 respectively. These changes were reflected in the $-\log_{10}$ adjusted p-values as well; the bulk of the IPW p-values did not change by more than a magnitude of 50 percent, but p-values for some estimates changed by more than a magnitude of 300 percent. While the effect on the majority of the IPW estimates, standard errors, and p-values was small, some of the top 50 genes saw substantial changes.

For the second sensitivity analysis where the observation with largest leverage value was deleted, the g-formula and IPW estimates were affected similarly ($(-8,6)$ percent difference) and to a slightly larger degree than the regression estimates ($(-2,3)$ percent difference). The change in standard errors was largest again for IPW but still contained to $(-1,5)$ percent difference, much smaller than for the previous sensitivity analysis. The change in g-formula standard errors was contained to $(-1,2)$ percent difference, and the regression standard errors changed by less than one percent in either direction. These changes were reflected in the $-\log_{10}$ adjusted p-values as well, with the percent difference for the regression and g-formula p-values having a spread comparable to the previous sensitivity analysis, and with the IPW p-values being considerably less variable than before.

Sensitivity Analysis 1: Delete Observation with Largest IP Weight



(a) Percent Difference in Estimates

(b) Percent Difference in Standard Errors

(c) Percent Difference in $-\log_{10}$ adjusted p-value

Sensitivity Analysis 2: Delete Observation with Largest Leverage Value

(d) Percent Difference in Estimates

(e) Percent Difference in Standard Errors

(f) Percent Difference in $-\log_{10}$ adjusted p-value

Supplementary Materials, Figure S.2: (a)-(c): Comparison of METSIM primary and sensitivity analysis results when deleting observation with largest weight. Top 50 genes are represented, ranked by p-value. (a), (b), and (c) respectively show the percent difference (primary - sensitivity) of the effect estimates, standard errors, and $-\log_{10}$ Benjamini-Hochberg adjusted p-values. (d)-(f): Comparison of METSIM primary and sensitivity analysis results when deleting observation with largest leverage value, which was the same observation for both the regression and g-formula methods. Top 50 genes are represented, ranked by p-value. (d), (e), and (f) respectively show the percent difference (primary - sensitivity) of the effect estimates, standard errors, and $-\log_{10}$ Benjamini-Hochberg adjusted p-values. R = Regression, W = Inverse Probability Weighting, G = Parametric G-Formula.